The Archive on Top of Your Desk: An Introduction to Self-Documenting Image Files [1993]

Thaller, Manfred

Historical Social Research
Historische Sozialforschung

Manfred Thaller:

The Archive on Top of your Desk. An Introduction to Self-Documenting Image Files [1993]

doi: 10.12759/hsr.suppl.29.2017.243-259

Published in:

Historical Social Research Supplement 29 (2017)

Cite as:

Manfred Thaller. 2017. The Archive on Top of your Desk. An Introduction to Self-Documenting Image Files [1993]. *Historical Social Research* Supplement 29: 243-259. doi: 10.12759/hsr.suppl.29.2017.243-259.

Historical Social Research
Historische Sozialforschung

Other articles published in this Supplement:

The Archive on Top of your Desk. An Introduction to Self-Documenting Image Files [1993]

Manfred Thaller[*]

Abstract: *»Das Archiv auf Deinem Schreibtisch. Eine Einführung in selbstdokumentierende Bilddateien«.* Recent technical developments made it realistic that individual historians could have work stations on their desks, which would allow them access to multi-GB collections of archival documents in digitized form. The problem with such a collection is, that the information systems of different archives usually handle their archival material in such a way, that it becomes very hard to extract content from one of them and integrate them into a local information system optimized to support the research interests of a specific user. An architecture for information system is described, which maximizes the possibility to transfer documents between different such systems. This architecture is based upon the concept of digital objects representing individual documents which contain all the semantic information necessary to integrate them seamlessly into different information systems.

Keywords: Manuscript processing, digital libraries, autonomous digital objects.

1. A Bit of Context

In 1989/1990 the author started to discuss publicly the concept of a Historical Workstation. It was basically described as a solution where a "powerful hardware platform" would be equipped with (a) software engineered to handle historical data, (b) read-only databases which provided access to "huge" amounts of machine-readable source material and (c) knowledge representations, which made the knowledge contained in traditional historical manuals available to a historian via appropriate software tools. In oral presentations the "powerful hardware platform" has usually been described as having one Gigabyte of disk storage plus high resolution graphics. At that time the notion that the typical historian might require one Gigabyte of hard disk seemed to be so outrageous to many colleagues, that this figure was usually dropped from printed presentations of the concept, being intentionally vague about the precise parameters of a "powerful hardware platform".[1]

[*] Reprint of: Manfred Thaller. 1993. The Archive on Top of your Desk. An Introduction to Self-Documenting Image Files. In *Image Processing in History: towards Open Systems,* eds. Jurij Fikfak and Gerhard Jaritz (= Halbgraue Reihe zur Historischen Fachinformatik A 16), 21-44. St. Katharinen: Scripta Mercaturae.

[1] E.g. in two (almost identical) introductions to collections of papers on the concept: Manfred Thaller, "The Historical Workstation Project", Historical Social Research 16 (1991) 4: 51-61,

At the time of printing (i.e. 1993), the author uses a cluster of workstations and PCs where he has access to approx. 7 Gigabyte of hard disk – of which, by the way approx. 1 Gigabyte is simply taken up by the operating systems and their swap space.

Until fairly recently the author has argued, that CD-ROMs are of only limited usefulness for the historian: as producing them required a minimal number of a couple of hundred copies to make the effort economically feasible, needing a fairly large investment, though the individual disk than might become cheap, it was, according to his opinion, much more sensible to concentrate on various forms of WORM techniques, where the price of an individual multi-megabyte disk would come down to about 100 DM plus local overheads.

At the time of printing the author has to pay about 30 DM to produce an individual multi session CD-ROM at the academic computing centre to which he has access.

Not all the revolutions which have been announced by the computer industry have taken place: the storage revolution, bringing real mass-storage to the desktop of an historian is in its hot phase. What does this revolution mean?

As always, there is of course more than one answer. There is one aspect of computer based work, however, where the accessibility of multi-Gigabyte storage devices makes a more fundamental difference than in most others. The subject of this volume: image processing.

Hardware has great promises for the further development of applications, but the promises have to be kept by software solutions embedded into an organisational framework: and that is the subject of this paper. For a start we focus in our discussion upon the handling of a very peculiar type of "image", that of a scanned manuscript. We do so, because the organisation of traditional archives is well understood by every historian, while the peculiarities of administering collections of images (pictorial sources) are not so widely known. Towards the end of the paper we will try to connect back to the applicability of the discussed techniques to such collections.

2. The Grail: Having an Archive on your Desktop

The author is aware of a far spread feeling that it will take a very long time until historians can expect to afford workstations. In the issue of Byte from August 1993 the new Power PC, which is being co-engineered by IBM, Apple and Motorola has been discussed at quite some length.[2]

---

doi: 10.12759/hsr.16.1991.4.51-61; Manfred Thaller, "The Historical Workstation Project", Histoire et Informatique, ed. by Josef Smets, Montpellier 1992, 251-260.

[2] Tom Thompson, Power PC Performs for Less, Byte August 1993, 56-74. The most important aspects in our context: (a) This system, which will support the full-fledged UNIX environment which is currently necessary to support the κλειω image analysis system mentioned by G. Jaritz and D. Buzzetti elsewhere in this volume [i.e.: G. Jaritz: Scratched Images or: Instead of an Introduction", in: In Jurij Fikfak and Gerhart Jaritz (Eds.): Image Processing in

Most notably in Sevilla, in the context of the project of the Archivo General de Indias[3], some historical archives have in recent years started to convert huge amounts of manuscript material into digitalised form. This author is aware that there exists a – rightful scepticism about information systems being built for unspecific purposes, as expressed e.g. by Ronald Stenvert:

> It will always be necessary to thoroughly consider why there is a need at all for an information system and if it is really all that necessary to store all sorts of information in the computer in a very broad purposed way.[4]

To a reader starting with this assumption the following paper may very much look like an exercise in "building information systems for the sake of information systems". Two arguments to the contrary: (a) Archival information systems will be built, not because of any pressure created by the community of historians, but because the make sense from a conservationist perspective. (b) If beyond that they shall make sense for historical researchers as well, it needs to be discussed what implications these needs should have for their implementation beyond the primary conservationist purpose. Let us summarize the specific reasons, which have generally been given for such projects:

To convert archival material into high quality digital images allows one to avoid accessing the originals thereafter. It therefore saves the archival material. (Conservation)

Processing such material in an environment where tools for image enhancement are available provides a much better access to the individual manuscript page than would be possible on a standard desk in the user room of an archive. (Manageability)

Storing the manuscript material in a central database system allows an institution to provide a copy "instantly" on the screen, without long waiting periods until the original is retrieved from a physical repository. (Accessibility)

Individual manuscript pages can quite easily be copied for an individual user, without threatening the original by physical damages, as each individual xeroxing operation does. (Reproducibility)

If a machine-readable description is attached to the manuscript, it is possible to look for the material much quicker than by traditional catalogues and *repertoria*. (Searchability)

If we look at these arguments, we discover a certain order in them: the conservation primarily interests the institution of the archive, the possibility to search for the document by its contents is probably the closest to the interest of the historical "end user", who wants casually to inspect a document and would be quite happy never to

---

History: towards Open Systems (= Halbgraue Reihe zur Historischen Fachinformatik A 16). St. Katharinen: Scripta Mercaturae, p. 9-20 and D. Buzzetti: Image Processing and the Study of Manuscript Textual Traditions, as above, p. 45-63.], uses a processor which costs about 50 % of the price of Intel's new Pentium (p. 64). (b) A version of the processor dedicated to notebook workstations has been announced for mid-1994 (p. 62).

[3] Cf. Pedro González: "The Digital Processing of Images in Archives and Libraries", in: Manfred Thaller (Ed.) Images and Manuscripts in Historical Computing (= Halbgraue Reihe zur Historischen Fachinformatik Vol. A 14), St Katharinen, 1992, 97-121.

[4] Ronald Stenvert Constructing the Past, Utrecht, 1991, p. 399.

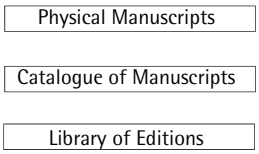
enter the archive, provided he or she can find what they are looking for in a printed edition. The intermediate arguments are of most concern for the non-casual user, for whom a specific document is so central that it is not possible to rely on an edition – or who, indeed, wants to edit the document him – or herself.

Anybody with archival experience will at this stage discover one or two differences between what the classical situation of an editor would be. Obviously the reproducibility of high quality digital representations of sources makes the editorial work much easier. But, on the other hand, the very fact that an image is reproducible, also means, that it is not automatically exclusively available for one researcher: so the role of the editor, who more or less keeps the archival material under his control until the edition is released, changes. On the other hand: if the archival material is reproducible at a quality which allows editorial work, why should that work take place in the archive at all? If, as stated initially, an individual CD-ROM may come down as low as 30 DM for, say 50 manuscript pages at 10 Megabyte each, which than would go way beyond the quality of most other reproduction techniques, why not simply have ten of them produced to bring a whole manuscript to your desktop?

There might be one reason, why this would not work immediately. Our fifth reason for digitalizing manuscripts has been the possibility to search for manuscripts with a specific content. In principle this means, that the manuscript as a whole is stored as a transcription as well, plus some additional information on the origin of it and similar information. What would this mean? Transcriptions of sources are usually provided by editions; and an edition provides beside it a description of the manuscript – which is the most simple component of a critical apparatus. By entering this kind of information into the medium by which a reproduction of the manuscript is made accessible, the difference between simple repository and edition gets blurred.

Before we take this argument further, let us compare the structure of a traditional archive and that of an archive providing part of its holdings with the help of an image data base with the help of a few diagrams. Diagram 1 shall visualize the traditional layout of an archive. The manuscripts are stored in a repository, they are described very summarily in a catalogue of manuscripts and the individual manuscript may be represented by an edition in the library attached to the archive.

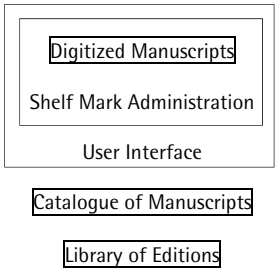
Diagram 1: Components of an Archive



If we decide to digitalize the images, we would in the diagram simply replace the repository of the physical manuscripts by an image database. If we do so, we need more than an unordered number of digitized images, they must be organized by some database structure. For the very goal of searchability that database would have to have some mechanism which allows the user to access specific images with the

help of at the very least an electronically administered list of the shelfmarks of the manuscripts. So we get diagram 2.

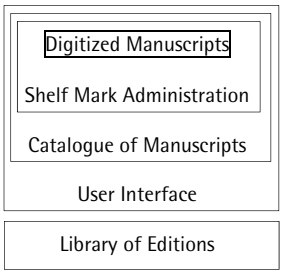
Diagram 2: Components of a Digitized Archive



What is a catalogue of manuscripts else than a paper-based data base documenting which manuscript is stored under which shelf mark? We arrived at diagram 2, because we wanted not to have thousands of images distributed randomly across our storage media, but we needed some principle for their organisation. As we see, the traditional way of accessing a manuscript is quite similar to the structure we arrived at on the computer: so it would be very sensible, to allow our user to interact not with the list of shelf marks, but with a database which represents the traditional catalogue of manuscripts and let the program which queries this catalogue interface into the ordering mechanism of the actual repository of digitized images somewhere in the background – which leaves us with the situation described in diagram 3.

This diagram is very similar to a technical description of the structure of current image databases: we will look into some components of this mechanism in more detail in a short while. Just to make our argument more transparent, let us reexamine, how such a system does actually work in a real life implementation:

Diagram 3: Integrating the Manuscript Catalogue



a) The interested historian gets access to the user interface of a database holding archival (or other pictorial) material.
b) The underlying DBMS administers primarily a rather straightforward database, which contains (usually highly structured) descriptions of the manuscripts in question.

c) This database in turn uses some addressing scheme (in the most trivial case simply the file names of individual image files) which is able to access a digital image of the manuscript.
d) This in turn resides in a repository: a collection of large disks, probably WORMs or CD-ROMs, which in the very few extremely well founded institutions will be handled collectively by a juke box for the appropriate media type.

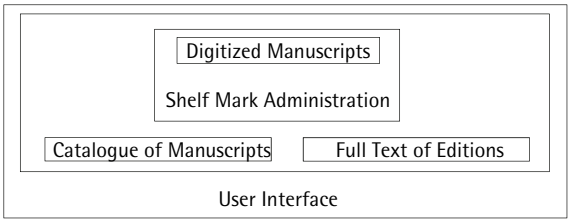
In our diagrams there remains so far one component, which is strangely unrelated to the unified database driven administration of the archival material: the library of editions, presumably physically present in the reference rooms of the archive. This describes the current situation quite well, where the books are indeed kept separate from the "real" archive, but in principle it would not have to be so. We will discuss the implications a bit later; let us just assume for the time being, that in the future editions will be produced by text processing systems and will therefore exist on electronic media. Data bases handling structured material and full text retrieval systems are generally quite separate pieces of software, but hybrid solutions between these two worlds exist. So there is no real reason, why the transcribed texts could not be integrated into our diagram as well, describing a situation, where the historian looks for specific combinations of phrases with the help of a fulltext retrieval system and instructs the system afterwards to retrieve the image of the manuscript page which has been transcribed in the edition. For a first attempt at realisation this would basically require, that the full text components of the overall system have themselves access to the same mechanism by which the individual manuscripts are accessed in the repository of digitized material as the structured components, administering the catalogue of manuscripts, have. We visualize this situation in diagram 4.

Before we continue further, let us look at the type of working process, which should be supported by such a system. We assume that a historian using the archive has access to a terminal. There he or she has the ability to browse (a) either through a computer based catalogue describing the holdings of the archive or (b) to create by various fulltext searches complex criteria which qualify a subset of the documents for display. The selected documents can then be displayed on the screen, enhanced, if necessary and presumably be printed.

It should be emphasized, that by our proposal to allow full text searches of transcriptions, we have changed the nature of the archival working process considerably. By accessing the content of the documents rather than their descriptions, the role of the archive has changed: it is no longer a repository for documents which have to be deciphered before they can be used. If such transcriptions exist: would not very many situations be imaginable, where the user of the archive will actually avoid accessing a manuscript in a difficult handwriting and simply use the transcription instead? The vast majority of the historians would access the originals only in cases, where the transcriptions indicate very doubtful readings. The reason, why we propose the integration of such transcription into the overall system is not that we think, that with high-quality editions, being the final result of a long drawn out editorial process, it will frequently be necessary to have access to the original documents. The reason is rather, that when a transcription has access to the transcribed original, it will become useable at a much earlier stage. Most historians will consider an edition of doubtful value, if it transcribes only the clearly readable portions and leaves all parts of the manuscript marked as doubtful where difficulties exist.

When we consider a transcription as a tool, which provides potential access to a high quality representation of the original, however, the situation changes quite radically: even a poor transcription, indeed even a revival of the classical *regesta*, could radically reduce the number of documents which have to be inspected in the manuscript or its digitized representation.

Diagram 4: Integrating Full Text Capabilities

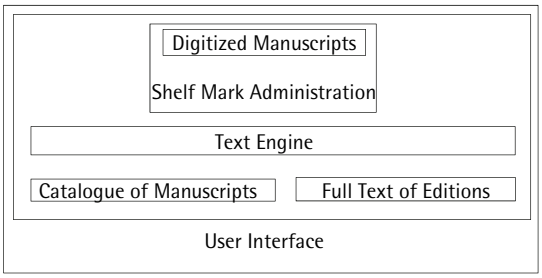


A close connection between digitally stored documents and transcriptions of them could therefore speed the access to historical documents up in more than one way: much material, which would never merit the fully drawn out effort of an edition, could become accessible considerably faster by providing preliminary transcriptions than we would think it possible today. Indeed, the whole notion of an edition should change in such an environment. At the moment a historical source is either edited or it is not. In the environment we describe here, many intermediate stages would be feasible: starting with a set of manuscripts which are accessible exclusively by their shelf mark; going on with manuscripts where tentative *regesta* exist as access filters; continuing with manuscripts, where a highly preliminary transcription exists; followed by manuscripts, where a mechanism has been implemented to point at parallel versions of the same section in another witness of the same base text; right through to a set of texts, where for the transcription a mechanism of annotations exist, which realize the potential of the various forms of apparatus, which make up the full power of a traditional edition as a tool of research.

Once again, by discussing the implications of a technical model for research, we have crossed a line, where this technical model has to be augmented. Our proposal to consider the "full text components" of our archival database as a kind of evolving edition, which is accessible to the historian already while it is being edited, forces us to the request a much more finely calibrated access mechanism to the manuscript material than so far. In a practical implementation this shelf mark administration mechanism which we postulated would in any case have to go quite a bit beyond the extremely simple idea from which we started. We said initially, that in its most trivial form it would require the catalogue of manuscripts to refer to, the names of the files in which the digitized material is stored. We left it like that to avoid the introduction of additional complexity at an early stage: but the assumption, that one entry in the catalogue of manuscripts would refer in a clear-cut one-to-one relationship to exactly one digitized file, has of course been an oversimplification. When we introduced a separate mechanism for the administration of shelf marks into our diagrams, we did assume, that such a mechanism would be able to resolve the reference to one item in the catalogue into the names of as many image files as

would be necessary to reproduce the archival entity to which the item in the catalogue refers: most typically probably one image file per manuscript page.

Diagram 5: Integrating Addressing within Images



The last refinement we introduced creates the necessity for a refinement in the opposite direction: while an item in a catalogue will frequently refer to more than one page, the integration of a functional equivalence for a critical apparatus into the full text database requires the possibility to refer to parts of a page as being the equivalent of a part of the transcription. We would, therefore, need a much closer link between individual fields in the structured parts of our access mechanism and portions of a running text on the one hand and segments of individual image files on the other. Therefore both, the catalogue of manuscripts (e.g. to show the incipit on the screen) and the full text components would have to access a nonlinear mixture between portions of ASCII text and parts of the bitmaps which reside in the background of the digital archive. Such a mechanism, a text engine, has already been described by the author elsewhere in this series.[5] Schematically it is reflected by the changes which are shown in diagram 5.

To have such an archive as an institution may sound Utopian today. But, if we assume that storage capacities continue to become cheaper at anywhere near the rate they did during the last five years in the near future, all preconditions for such designs would be available. Let us bring our user-oriented considerations one step further, though. When central archival resources, like the manuscripts themselves, their catalogues and even steps towards editions of the manuscripts, would be machine readable, why would it still be necessary to do all the work at the physical location of an archive? We mentioned initially, that the production costs for individual CD-ROMs have become astonishingly cheap recently. So, why should one not transfer substantial portions of an archive to one's own desk and do the work there, where all the other tools necessary for the study to be undertaken would be available?

The attractiveness of the hypothetical system we described above lies in its integration: in such an environment a historian using the system could draw from many resources, which are currently not easily combined. Precisely this high degree of

[5] M. Thaller, "The Processing of Manuscripts", in: Manfred Thaller (Ed.) Images and Manuscripts in Historical Computing (= Halbgraue Reihe zur Historischen Fachinformatik Vol. A 14), St. Katharinen, 1992, 97-121.

integration makes "moving the archive to your desktop", however, a very difficult operation. Copying individual image files would of course be possible quite easily: but just about all the comfort we have described on the preceding pages would be lost. The historian doing so would end up not with the archive on his or her desktop, but with a (potentially) large number of unrelated files. The only solution would be to copy virtually the whole data of the archive – and while storage media are getting more ample, there is some reason to believe, that central installations will remain more amply provided with storage capacity than individual researchers. The design we have described so far, makes it extremely cumbersome, however, to export a subset of the data administered. These difficulties would become even more severe, when we would realistically assume, that most historians would like to have on their desks not just material from one archive, but from a number of such. If we do not assume that all archives would adopt one and the same solution for the realization of such a conceptual model, the connections between images of manuscript pages, short descriptions and partial transcriptions would be realized in vastly different ways.

3. Tools for a solution: Self Documenting Image Files

In the remainder of this paper we would like to describe a solution which should make the following possible.

1. A historian should be able to use whatsoever system a specific archive provides to select out of large holdings those portions of the material – be it manuscripts or images – which are relevant for him or her.
2. These materials should exist in a form, which supports their export from the individual institutional databases.
3. And on local machines the possibility should exist to recombine such files from different institutions rapidly into consistent local databases, which still implement the central characteristics we arrived at in the preceding section.

It may be necessary to point out at least one problem which we did not mention in the preceding paragraphs. Many readers might assume, that the access to an archival institution of the type described here will in the future be organized via network. This may very well be so – we would claim, however, that this is only a partial solution. Just to give an impression of the magnitude of the problem: this booklet as a whole, when transmitted as ASCII text, will need approximately one fifth of the storage space required, which a digital image of a medieval charter needs – at least if it is good enough for palaeographic considerations. So downloading image files will in all probability remain for quite some time to come the only reasonable use made of networks in image related work, rather than working with the images on line via a network. In computer science, one hesitates to quote a book which is thirteen years old. Still, the following sentence on situations where data bases should not be located at remote sites, is to the best knowledge of the author as true as ever:

> Another problem characteristic is where a node makes infrequent but very data-intensive access to a file. No distributed system, no matter how cleverly designed, could perform well an application like this.[6]

This almost reads like an intentional description of archival databases. Downloading an image – of a manuscript or a carefully described pictorial source – together with all the information connected to it is, however, precisely the kind of process we have in mind here, when we discuss reconstructing the functionality of an archive system with a smaller amount of data on a local workstation.

To make the problem we deal with more explicit: according to the design we developed in the preceding section, we have the following types of information which are related to one manuscript page or the digital representation of an image:

1) The digital image itself.
2) Cataloguing information, as a set of fields like they occur in structured databases.
3) Parts of the full text contained in the manuscript page or full text descriptions of portions of an image, where a structured description would not work
4) Links between parts of the information described under 2) and 3) and subareas of the image itself or potentially other images which may or may not have been downloaded as well.

At the most trivial this means, that we would for each individual image receive at least two, probably more, physical files, the relationship between which should be correctly re-established after downloading.

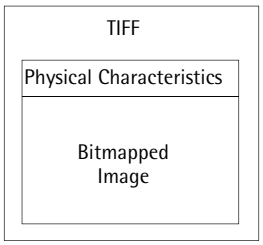
A similar problem exists for quite some time already in the handling of images. Image files contain basically a bit map; that is a long string of bytes (or multiples of bytes) each of which describes an individual pixel of the image. Besides this bit map, however, there are at the very least two more items required to interpret an image correctly: its height and its width. (A 12 pixel image otherwise could be interpreted as a 2 by 6, 3 by 4, 4 by 3 or 6 by 2 image.) This information about an image is usually described as part of the physical characteristics of an image, together with information on how many bits per pixel are actually used, whether a compression algorithm has been used, which one it has been and which 256 colours to select out of the millions which can be displayed on a modern screen shall be used. (On equipment which altogether can display millions of potential colours, quite frequently only 256 can be displayed concurrently.)

Older solutions to this problem provided for a header within an image file: the first n bytes would provide the necessary information in a fixed order. More modern solutions, like the widely used TIF Format (Tagged Image File Format), acknowledge that the possibilities to store transmit and display an image are so variable, that two images will typically have to be described with completely different characteristics. The provide a mechanism, therefore, by which such a header has a variable length and describes such characteristics out of potentially extremely many, as are needed to describe specific images. A schematic representation of this is provided in diagram 6.

---

[6] G. A. Champine, Distributed Computer Systems, Amsterdam etc.: North Holland, 1980, 63.

Diagram 6: Contents of a TIFF-File



In theory, there would, of course be the possibility to handle these two parts of the file in different physical files: indeed, that would in some situations be more useful, as it would speed the translation process between different image file formats considerably, when a constant file, representing the actual bitmaps, could be processed together with independent descriptions of the contents of that first file. For all practical purposes the minor gain in flexibility, which would be provided by a separation of physical description and actual bitmap, is considered to be more than compensated by the negative effects of having to keep track of which bitmap file would be described by which physical description.

The obvious solution for our problem above – how to export data from an archive containing digitized manuscripts or pictorial sources together with descriptions – should therefore be an extension of the principle we just discussed, an integration of these historical descriptions of the meaning of an image, as well as the technical description of its physical properties. Indeed, the more flexible image formats of today, like the TIFF mentioned above, lend themselves rather easily to the introduction of additional items of description. Producing such "modified TIFF" headers has a risk: it may preclude processing the image by other software. But, interestingly, this may not be an unwelcome side effect in an archival environment, as it makes control over what the end user does with the material (and control of implied copyright issues) much more feasible.

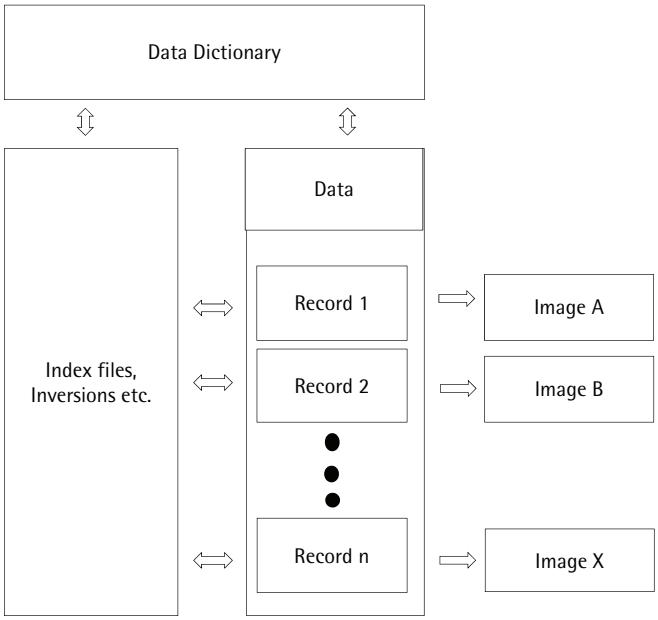
We cannot leave it at that, however, as the practical implementation of such a design involves the solution of decidedly non-trivial problems. These problems fall into two categories: (a) on the one hand, as we have said before, extricating information from an integrated system as the one we described before can be rather complex; (b) on the other, the transfer of historical descriptions of source material – be it pictorial or manuscript-like – will raise again all the questions which have been encountered during the standardization discussions of the last ten years now.

Why is the extrication of information from a database system like the one which we have described so complex? To explain the issues involved, we would have to look more careful on the way in which such systems are actually implemented. We have given in diagram 5 a description of the functional layers of an integrated system from a user's point of view. Diagram 7 attempts to show the internal structure of a database system that could provide such functionality. This diagram is to be read as follows:

To allow the user to interact with a database, the DBMS administering it has to have some knowledge about the data. With the exception of some extremely primi-

tive systems, a DBMS can handle more than one database: so a list has to be administered, which describes the fields contained in the database, the data type these fields have, whether a specific field has been "inverted" and so on.

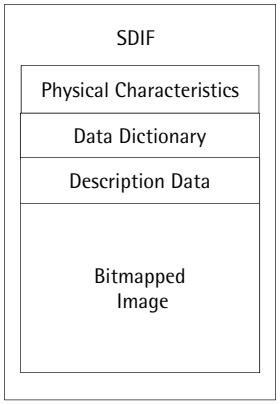
Diagram 7: Traditional Architecture of an Image Data Base



This type of information appears in our diagram as the data dictionary of a traditional image database design. (Here as in all the following items we do not discuss, whether such a data dictionary is realized as one or more actual physical files.) The data in a database, the central rectangle in the diagram, is made up by the actual descriptions of an image. For our present purposes we ignore the differences between full text and structured systems at the moment. The division of the data into "records" which are contained within a bounding rectangle, are an indication that in many cases the descriptions applied to individual images are just abstractions which are meaningful only, if accessed in the context of the specific DBMS: the description of a specific image appears as one distinct record when looked at by the user. In many database designs, notably in all relational ones, the actual data making up such a logically consistent description, will, however, be distributed over a number of physical files. They can be handled meaningfully only within the environment of the respective DBMS, therefore. With index files, inversions, etc. we try to introduce all components of database systems into our diagram, which are derived by the system from the contents of the data section, being organized into such sets of additional independent files as necessary, but not containing any information which could not be reconstructed from the data section of the database. Practically such components are introduced into the overall layout of databases primarily by a con-

cern for speed. User queries are handled by interactions between these three components: the data dictionary contains the knowledge which is necessary to transform the query of a user into meaningful navigational operations within the data portion or the available index files and similar tools. Usually an index file is just a kind of lookup table for the data section. It generally consists of keys which are connected with an address within the data section: for example that portion within a full text system which contains a specific term indexed for quick access. (Strictly speaking, therefore, in most cases there is only the possibility to use the index to look something up in the data section. As some advanced designs exist, however, where there are also possibilities to go from the data section implicitly to an index to find "similar documents / records", we have drawn the arrows headed in both directions.) As images have been added to almost all databases only as a kind of afterthought, the rightfully exist in the diagram only as quite badly integrated boxes at the right side.

Diagram 8: Selfdocumenting Image File (SDIF)



Almost all existing image databases handle their images in exactly that way: an image is represented by a file name in the database and while there is a connection between database and image file, looking at an image file you have no way to find out to what data in the database this image is connected. The implication of a successful conclusion of the research done about "unique identifiers" for images by William Vaughan and others would be, that information drawn from the images would also lead to the creation of access mechanisms as provided by index files for textual information in a data base. No reliable architecture for such purposes is known to the author, though.[7]

To package the information from such a design into one physical file, which contains – independently from any other file – all the information related to the

[7] On "unique identifiers" for images cf. W. Vaughan "Paintings by Number: Art History and the Digital Image", in: A. Hamber et al. (Eds.): Computers and the History of Art, London and New York: Mansell, 1989, 74-97.

description of an image and all the information needed to process these descriptions, we have to extract some portion of the data dictionary as well as parts of the actual data section. As such a file, schematically presented in diagram 8, contains all the information necessary to understand the description of the image contained within it, we call it a self-describing image file or SDIF. (Of course a researcher has to prepare the historical description first: by self-describing we mean, that not only the bit map and not only the text of a description is contained in the file, but also a "description of the description" specifying which fields are actually used and so forth.)

Our reasoning so far could be summarized as follows: Image databases will increasingly become important for archival institutions of both, the traditional kind handling manuscripts and the more modern variety, administering pictorial sources. To export material from such databases for use on the desk of individual historians, we should work for an exchange format, which packages descriptions of images and the images themselves together into one file. Such a file we call a self documenting image file.

But there are further implications of this concept. It has been mentioned above already, that using information out of various archives at the same time, would introduce a whole set of additional complications in the area of standardizing the images. We have not described yet, what happens to a SDIF at the receiving end, on the desktop of the historian who wants to analyze the material. In principle there is no reason, why a SDIF should not be "unpackaged" and re-integrated into a database of the design we have presented in diagram 7. There is one problem, however: such databases do assume that the data a database contains has basically one consistent format. This leaves us with two options: either to find a definite prescriptive standard for all databases which in the future will contain historical source material. This author has already commented elsewhere, why he considers such prescriptive standards as a logical impossibility, leaving aside the question whether they are organizationally viable – and not even mentioning the question, how encompassing such a standard could become.[8]

The alternative would be to look for a database design, which allows for databases where individual parts of the data have quite different structures. Such an approach would require to reconsider some of the basic decisions of existing database architectures.

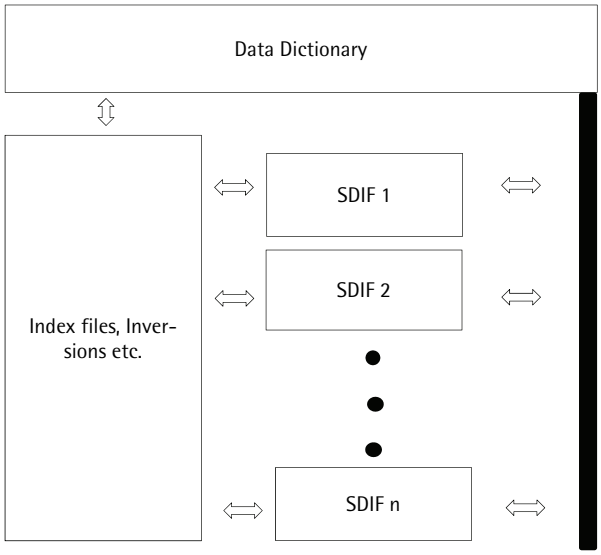
What is a database? Looking into classical text books, we get definitions like "A database system is essentially nothing more than a computerized record-keeping system. The database itself can be regarded as a kind of electronic filing cabinet – that is, as a repository for a collection of computerized data files."[9] or "A database can be defined as a computerized collection of stored operational data that serves the needs of multiple users within one or more organizations. (...) The database is not single-program oriented as were private data files, but has an integrated re-

[8] M. Thaller, "The Need for Standards: Data Modelling and Exchange", in: Daniel Greenstein (Ed.), Modelling Historical Data, (= Halbgraue Reihe zur Historischen Fachinformatik Vol. A 11), St. Katharinen, 1992, 1-18.

[9] C. J. Date, An Introduction to Database Systems, vol. I, Reading etc.: Addison-Wesley, 4th ed. 1986, 3.

quirements orientation."[10] These definitions are abstract: all they require of a database is, that it has the ability to provide users with integrated access to a body of data. There is nothing in these definitions, which makes any prescriptions about the way in which data have to be distributed between individual files. Indeed, as far as the text book definitions are concerned, a database could also be implemented as a set of very many extremely small files up to the point where each file contains just one field. That such an implementation will never occur, is not based in theory, but in low level characteristics of computer systems: very small files waste disk space, as each file has a small overhead (typically using actually as much space as to round the file size up to the next multiple of 512 or a similar number); they are disastrously slow, as making an individual file ready for reading takes much longer than an actual reading operation. Therefore current database architectures tend to realize the rectangles labelled "data dictionary", "data" and "index files" in diagram 7 with the smallest possible number of physical files possible.

Diagram 9: Fragmented Architecture of an Image Data Base



As soon as we are processing images, however, quite a few of the basic assumptions change. Reading a high quality digital manuscript from its five to ten megabyte file into memory and displaying it on the screen is an operation besides which the act of opening the file loses almost all relevancy for considerations of speed and images, as a rule, are stored in separate individual files to begin with. We would therefore propose to look for databases which shall contain variously described

[10] Toby J. Teorey and James P. Fry, Design of Database Structures, Englewood Cliffs: Prentice-Hall, 1982, 4.

images according to completely different designs, presented in diagram 9, which for probably obvious reasons we call a fragmented database design.

This diagram should be read as follows: As in the traditional architecture all information about the internal structure of the database is kept in a data dictionary. Data and image portion of the database are, however, merged completely into a collection of SDIFs which remain physically separate files. The data dictionary still contains information to prepare navigational operations: that information consists, however, primarily of tables which describe in which SDIFs it will be sensible to look for which structures, while the individual SDIF is expected to contain all information necessary for all navigational operations within itself. (A mechanism which is supposed to be indicated by the black box at the right side of the diagram.) Speeding up the access uses the traditional mechanism: the only difference is, that we assume the addresses in the index files to be realized not as addresses within a number of files defined by the DBMS which represent the "data" section of diagram 7, but as addresses within individual SDIFs.

Such a design allows for the rapid integration of additional SDIFs into a local working environment. It makes no assumptions about the similarity of structures of individual SDIFs: as the descriptions of the images are kept separate, the may have completely separate logical structures. The question which remains is, how a meaningful navigation in such divergent structures should be realized. We propose to do so by the assumption that the names of individual entities and fields chosen by the users are semantically meaningful in as far, that it is guaranteed, that something which is called an "x" in two different SDIFs is the "same" kind of information. So extracting all "things which have a surname" should indeed produce a list of persons.

To guarantee this semantic consistency a number of strategies would be possible. One would, e.g., be to agree upon a fairly long list of potentially useful fields. Such fields would be related to an overall implementation of a database relying on "semantically meaningful names" exactly as the concept of semantic primitives is to the concept of semantic networks in the world of AI[11], so that the fields occurring in a given document – whatsoever their name in the original database – could reference such a meta terminology. Such fields could be combined to "entities" which could reference a similar list: an entity of type person, e.g., always having a minimal set of fields, which in actual implementations of databases could be augmented by additional ones. It is fairly obvious, that this concept is closely related to the notion of inheritance familiar from object oriented programming.[12] Outside a fairly specialized community it is less widely known, that under the heading subtypes and super types similar considerations have been introduced into the context of relational modelling already fifteen years ago.[13]

---

[11] On both see Avron Barr and Edward A. Feigenbaum, The Handbook of Artificial Intelligence, vol. I, Reading etc.: Addison-Wesley, 1989, pp. 207215 and 180-189 respectively.

[12] See e.g. B. Meyer, Object Oriented Software Construction, New York etc.: Prentice Hall, 1988, 217-280.

[13] E.F. Codd, "Extending the Database Relational Model to Capture More Meaning", in: ACM TODS 4, No. 4 (December 1974), or, easier to read: C.J. Date, An Introduction to Database Systems, vol. II, Reading etc.: Addison-Wesley, 1985, chapter 6 (= pp. 241-289) here: pp. 243

These questions – which have partially been discussed in other publications of this author[14], "Prototypes of Information contained in Historical Sources" go well beyond the scope of this paper, however. We end, therefore, by summarizing the second part of our argument for self documenting image files as: Beyond being a useful abstraction to handle the data transfer between traditionally designed data bases, SDIFs could furthermore become building blocks for a new database architecture, which we call fragmented. This architecture lends itself particularly easily to situations where highly divergent data structures have to be combined into one working environment. To realize it, we propose to require consistent naming conventions between individual database systems, when data from them are being exported for integration at other sites.

and 263-269. On the differences between the two see R. King, "My Cat Is Object-Oriented", in: Won Kim and Frederick H. Lochovsky (Eds.), Object-Oriented Concepts, Databases, and Applications, Reading etc.: Addison-Wesley, 1989, 23-30.

[14] Cf. M. Thaller, "A Draft Proposal for a Standard for the Coding of Machine Readable Sources", in: Daniel Greenstein (Ed.), Modelling Historical Data (= Halbgraue Reihe zur Historischen Fachinformatik Vol. A 11), St. Katharinen, 1992, 19-64; in the current context: pp. 54-64.